/// LCA 求两点的最近公共单亲节点

#define mem(a) memset(a,0,sizeof(a))

#define rep(i,a,b) for(int i = a; i<=b ;++i)

using namespace std;

const int maxn = 10005;

vector<int> edge[maxn];

int r[maxn], depth[maxn<<1], occur[maxn<<1], cnt;

bool in[maxn];int dp[maxn<<1][20];

void dfs(int u,int deep){

occur[++cnt] = u; //cnt次出现

depth[cnt] = deep;//深度

if( !r[u] )

r[u] = cnt; //occur

for(int i = 0; i < edge[u].size(); i++){

//rep(i, 0, edge[u].size()-1){

int v = edge[u][i];

dfs(v, deep+1);//深搜

occur[++cnt] = u;//回溯

depth[cnt] = deep;//回溯

}

}

void rmq(){

rep(i, 1, cnt)

dp[i][0] = i;

int nn = int(log((double)cnt)/log(2.0));

for(int j = 1; (j) <= nn; j++)

for(int i = 1; i+(1<<(j-1))-1 <= cnt; i++)

dp[i][j] =

depth [dp[i][j-1] ] < depth[ dp[i+(1<<(j-1))][j-1] ]

? dp[i][j-1] : dp[i+(1<<(j-1))][j-1];

}

int query(int x, int y){

int l = r[x], rr = r[y];

if( l > rr) swap(l , rr);

int k = int(log((double)(rr-l+1))/log(2.0));

int mmin =

depth[dp[l][k]] < depth[dp[rr-(1<<k) + 1][k] ]

? dp[l][k]: dp[rr-(1<<k) + 1][k];

return occur[mmin];

}

int main()

{

int t, n, u, v; scanf("%d", &t);

while( t-- ){

scanf("%d", &n);

rep(i, 1, n) edge[i].clear();

mem(in), mem(depth), mem(dp), mem(r);

rep(i, 1, n-1){

scanf("%d %d", &u, &v);

edge[u].push\_back(v);

in[v] = 1;

}

cnt = 0;

rep(i, 1, n){

if( !in[i] ){

dfs(i, 0);

break;

}

}

rmq();

int l, rr;

scanf("%d %d", &l, &rr);

printf("%d\n", query(l, rr));

}

}